

IN THE SPECIFICATION

Please amend the paragraph beginning on page 10, line 19 as follows:

In one embodiment, the system may also include an initial payload that is stored in Read Only Memory BIOS (ROM BIOS). In one embodiment, the initial payload is part of the first software module (e.g. the ISUA). In an alternative embodiment, the initial payload is stored as a module in ROM BIOS, separate from the first software module. In one embodiment, the initial payload is launched from ROM BIOS and displayed on the screen after the Power On Self Test (POST) but prior to the booting, loading and/or execution of the OS. This may occur at a predetermined time, such as when the system is being manufactured, assembled and tested, or when the end user first activates the system. In an alternate embodiment, this initial payload is copied to a predetermined location (such as the system's hard disk) at a predetermined time, such as when the system is being manufactured, assembled and tested, or when the end user first activates the system.

Once copied, the payload executes after POST but prior to operation of the OS, and may display graphics, advertisements, animation, Joint Photographic Experts Group (JPEG)/Moving Picture Experts Group (MPEG) formatted material on the screen. When additional programs and/or payloads are delivered (via the Internet or other outside connection), the display screen may be used to provide customized screens in the form of messages or graphics prior to and during booting of the OS. In addition, executable programs delivered in the first software module, as well as subsequent programs (such as the second software module) downloaded from the web site, may be used to survey the PC to determine various types of devices, drivers, and applications installed. In one embodiment, as described in co-pending U.S. Patent Application serial number [] 09/336,289, entitled "Method and Apparatus for Automatically Installing And Configuring Software on a Computer", filed June 18, 1999, assigned to Phoenix Technologies Ltd., the contents of which are incorporated herein by reference, the first software module is used to identify and to automatically create shortcuts and/or bookmarks for the user. The programs downloaded from the website may include software that collects and maintains a user profile based on the user's preferences. Such information may be provided to the infomediary, which subsequently forward portions of

the information and/or compiled data based on the information to suppliers and other business to obtain updated or revisions of information provided by the suppliers and other businesses.

Please amend the paragraph beginning on page 17, line 20 as follows:

--RAPI 84 generally provides a secured interface between ROM application programs and system BIOS 82. One embodiment of RAPI 84 is described below in Figures 8 through 18 and the accompanying text. One embodiment of IUSA 86 is described in co-pending U.S. Patent Application serial number 09/336,289, entitled "Method and Apparatus for Automatically Installing and Configuring Software on a Computer," filed June 18, 1999, assigned to Phoenix Technologies Ltd., and which is incorporated herein by reference.--

Please amend the paragraph beginning on page 18, line 1 as follows:

--One aspect of the present invention is described with reference to an operating system installed on the processing system 100, shown in Figure 2. Figure 4 is an overall functional block diagram illustrating the architecture of a processing system utilizing the system and method of the present invention. The processing system 100 comprises an operating system ~~2230~~ 230 which supports application programs 232 and services ~~2234~~ 234, Basic Input/Output System ("BIOS") 236 and system hardware 238. The BIOS 236 is a collection of drivers, or software interfaces for hardware devices such as the console (keyboard and display), a generic printer, the auxiliary device (serial port), the computer's clock and the boot disk device. The BIOS 236 is typically embedded in programmable, read only memory (ROM). Often, the BIOS functions themselves are actually copied from ROM into physical memory, taking advantage of the faster access times of physical memory. This is known as "shadowing" the BIOS 236 because two copies of BIOS 236 results: one in ROM (which will no longer be used) and the other in physical ~~memory~~ memory. The portion of physical memory which stores the BIOS 236 is known as the BIOS shadow space. An operating system such as Windows NT makes no use of the BIOS 236 after the operating system has been booted and is running. The kernel level drivers in the Windows NT operating system interface directly with the

system hardware. The present invention facilitates the use of the BIOS 236 as an interface between system hardware 238 and an operating system ~~232~~ 230.--

Please amend the paragraph beginning on page 18, line 22 as follows:

--The operating system 230 includes a class driver 240 which interfaces with the applications programs 232 and services ~~2234~~ 234, and an I/O Manager 242. The I/O Manager 242 converts I/O requests from the application programs 232 and services 234 (made via the class driver 240) into properly sequenced calls to various driver routines located in the kernel 244. In particular, when the I/O Manager 242 receives an I/O request, it uses the function codes of the request to call one of several dispatch routines in a driver located in the kernel 244. The kernel 244 provides hardware-independent functions, called system functions, that are accessed by means of a software interrupt. The functions provided by the kernel 244 include file and directory management, memory management, character device input/output and time and date support, among others. In one embodiment, the operating system is the Windows NT operating system. In alternate embodiments, the operating system 230 includes the Solaris or the AIX operating systems or other operating systems based on demand-paged virtual memory subsystems.--

Please amend the paragraph beginning on page 19, line 11 as follows:

--The present invention provides an access driver 246, located within the kernel 244, which is responsible for accessing BIOS data located in the BIOS 236 or for accessing system hardware 238 data via the BIOS 236. The access driver 246 is also responsible for accessing the location of a BIOS function address, as well as executing the associated BIOS function. In one preferred embodiment, the access driver ~~244~~ 246 comprises source code written in C language. It is understood that other assembly languages may be utilized in implementing the functions of the access driver ~~244~~ 246. The BIOS data and addresses are typically located in physical memory 250 and are accessed by the access driver 246 via a BIOS Interface 248. In one embodiment, the access driver 246 executes code in the BIOS shadow space, typically at physical addresses 0x000E0000 through 0x000FFFFF.--

Please amend the paragraph beginning on page 20, line 5 as follows:

--In one preferred embodiment, a set of entry-points or ~~functions~~ function calls are available to the application programs ~~230~~ 232, services ~~232~~ 234 or class driver 240 which utilize the access driver 246. The access driver 246 can be opened, closed, and can receive input/output ("I/O") control codes ("IOCTLs") through these entry points. Table 1 illustrates the structure, entry points and applications for the access driver 246.--